APPENDIX – E

RCP GATE EFFICIENCY AND LOAD BALANCING :

The theory behind Rcp gate efficiency is illustrated below :

To begin with it is important to note that the stage four of the Bind_Virtual_Queues function 3207, deals with three different variations of bind sequence numbers namely the next output bind sequence num field 2815, in the Rcp Gate node, and the next output bind sequence num field 1903, in the local ring node, and the output bind seq num 2906 in the bind node structure of the bind table entry.

The number of outputs alloacted by the Rcp gate, is stored in the next output bind sequence num field 2815 of the Rcp gate node structure. It may be noted that this value for the rcp gate only, where as the next output bind sequence num field 1903, in the local ring node, may contain a higher value and pertains to the number of outputs allocated by all rcp gates connected to the local ring. The output bind seq num 2906 in the bind node structure of the bind table entry, is an instance of the next output bind sequence num field 1903, in the local ring node.

It may be noted that we are interested in the next output bind sequence num field 2815 of the Rcp gate node structure, and this number represents number of outputs allocated by the rcp gate, and may be slightly more than the number of outputs processed, but serves as a good approximation.

The number of times a worker is assigned to the node function invocations of the rcp gate is stored in the num of worker assignments field 2806 of the Rcp

gate node structure. Since we know the sizes of the queue arrays of the Rcp gate, we can deduce the following,

a) The number of queues processed per each worker assignment, by dividing the output bind sequence num field 2815 by the number of worker assignments.
b) Dividing the number obtained above by the capacity of the queue arrays and multiplying by 100 gives the percentage of queues processed per each worker assignment, and is termed as the rcp gate efficiency.

The theory behind load balancing is a series of simple ideas, as explained below.

a) Node functions can have arbitrary loads which are unknown, however a node function invocation of heavier load will take longer time than a node function invocation of lesser load.
b) In view of the above, it can be stated that the input queues of a node function invocation of heavier load build up, whereas the input queues of a node function invocation of lesser load are rapidly consumed, hence they evaporate.
c) In addition, it can also be stated that a node function invocation of heavier load, keeps running for longer times, compared to a node function invocation of lesser load, which completes processing quickly.
d) In view of the above it can be stated, that the Rcp gate efficiency of the rcp gate controlling the node function invocations of lesser load, will be poor, as compared to the rcp gate efficiency of the rcp gate controlling the node function invocations of heavier load.
e) In view of the above, it can be deduced, that when a node function invocation is running, and if the Rcp gate efficiency is very high, and if

the inputs and outputs available are very high, then that node function invocation is of heavier load.

f) Similarly it can be deduced, that when none of the node function invocations of the rcp gate are running, , and if the Rcp gate efficiency is very low, and if the inputs and outputs available are very low, then the node function invocations of that rcp gate are of lesser load.

The Rcp architecture uses the same principles, and considers 75% of Rcp gate efficiency and 75% of input and output queues available, and one or more node function invocations running, as a sign of heavier load. Similarly, 25% of Rcp gate efficiency and 25% of input and output queues available, and none of the node function invocations running, as a sign of lesser load.

When heavier load is detected more node function invocations are dispatched if possible, and when lesser load is detected, the dispatching of node function invocations is temporarily bypassed until the percentage of input and output queues available is greater than 25%.

Automatic load balancing my be achieved by blindly setting node max invocations configuration parameter to 2 or more, for every node function in the application. The Rcp runtime will compute rcp gate efficiencies and will switch workers automatically towards node function invocations of heavier load.